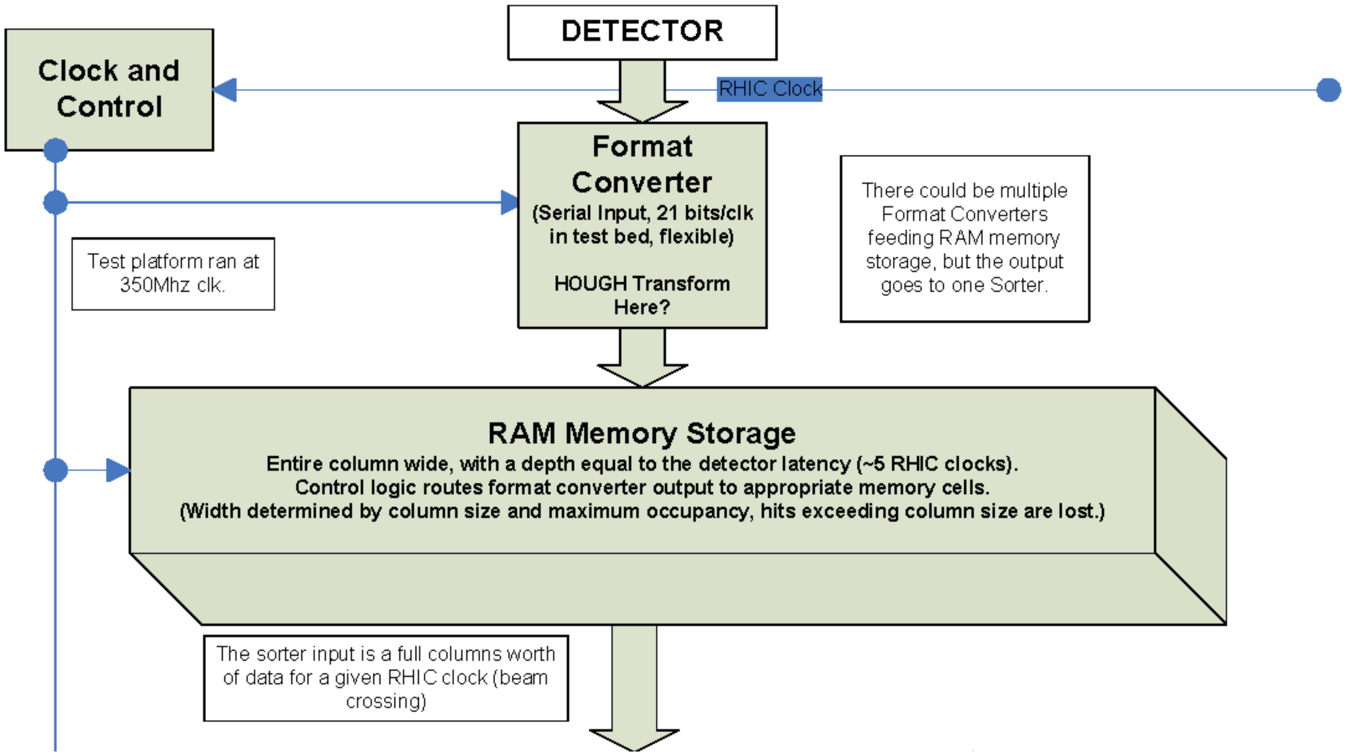
The FVTX LL1

- The FVTX LL1 started as a DOE STTR grant to ISU and Northern Microdesign
  - C. Ogilvie, J. Lajoie, W. Black, N. Badr
    - Phase-1 work completed 2004-2005
    - Proof-of-Principle for FPGA-based displaced vertex tracker
      - Zero magnetic field
      - Demonstrated event vertex finding, displaced vertex identification – all in FPGA hardware
  - Phase-II work starting now
    - Implementation in non-zero magnetic field
    - Interface to FVTX
      - Prototype to interface to LDRD tracker
    - Construction of prototype hardware

FVTX LL1 – Phase-1 Interface

- The Phase-1 interface was design to accommodate asynchronous input
  - This was beneficial to testing as well as being matched to the proposed FVTX hardware.


DETECTOR
Clock and Control
RHIC Clock
Format Converter
(Serial Input, 21 bits/clk in test bed, flexible)
HOUGH Transform Here?
RAM Memory Storage
Entire column wide, with a depth equal to the detector latency (~5 RHIC clocks).
Control logic routes format converter output to appropriate memory cells.
(Width determined by column size and maximum occupancy, hits exceeding column size are lost.)

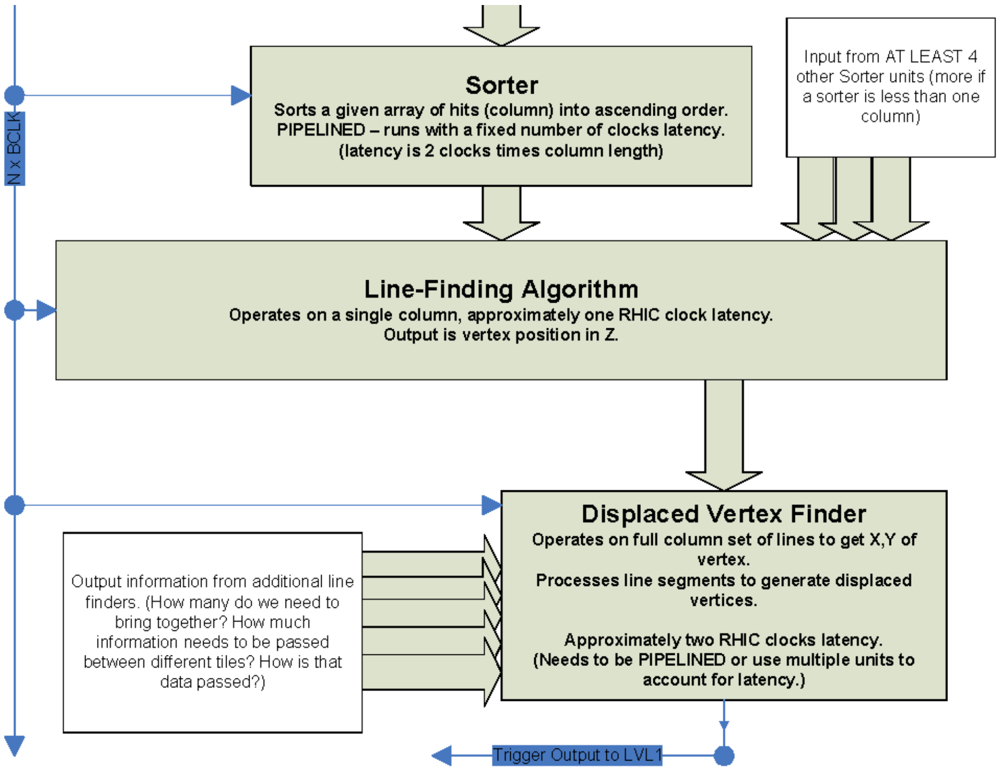
There could be multiple Format Converters feeding RAM memory storage, but the output goes to one Sorter.

Test platform ran at 350Mhz clk.

The sorter input is a full columns worth of data for a given RHIC clock (beam crossing)

Phase-I Algorithm

- The Phase-1 algorithm was a “brute force” method applied to finding line segments
  - Line segments yield event vertex
    - Sorting lines yields displaced vertices
  - This stage requires cross-stitching between FPGA boards
    - Still grappling with this problem – how complex will it be?



Sorter
Sorts a given array of hits (column) into ascending order.
PIPELINED – runs with a fixed number of clocks latency.
(latency is 2 clocks times column length)

Input from AT LEAST 4 other Sorter units (more if a sorter is less than one column)

Line-Finding Algorithm
Operates on a single column, approximately one RHIC clock latency.
Output is vertex position in Z.

Displaced Vertex Finder
Operates on full column set of lines to get X,Y of vertex.
Processes line segments to generate displaced vertices.

Approximately two RHIC clocks latency.
(Needs to be PIPELINED or use multiple units to account for latency.)

Output information from additional line finders. (How many do we need to bring together? How much information needs to be passed between different tiles? How is that data passed?)

Phase-II Interface

- From the FVTX proposal, data transfer to LL1 will go through the cumulator board.
  - The cumulator board will collect data and organize by beam crossing.
  - One 7.5Gbit fiber per octant
    - This means it will take 2-4 RHIC clocks to transfer the data from one beam crossing (at 1.5% occupancy for a AuAu event)
    - THIS WON'T WORK…
      - …unless you run multiple LL1's in a barrel-switch fashion
        - » EXPENSIVE!
- A better way (?)
  - Pass the data <u>through</u> the cumulator board
    - Take advantage of additional transfer time
    - Let LL1 organize and transform the data

Scope of FVTX LL1

- How large will FVTX LL1 be?
  - Planning on servicing ~180k channels with a single “tile” (FPGA module board)
    - This corresponds to 8 “wedges” across four stations
    - 32 fibers per tile
      - Similar in scope to what we did for MuID LL1
    - Six boards per endcap, twelve for both arms
- Phase-II prototyping
  - Expect that Phase-II hardware prototype will be able to service LDRD VTX detector
    - Keep same design spirit, modify input logic as necessary
    - Prototype hardware developed under Phase-II STTR